

A note on multipivot Quicksort

Vasileios Iliopoulos[‡]

July 29, 2014

Abstract

We analyse a generalisation of the Quicksort algorithm, where k uniformly at random chosen pivots are used for partitioning an array of n distinct keys. Specifically, the expected cost of this scheme is obtained, under the assumption of linearity of the cost needed for the partition process. The integration constants of the expected cost are computed using Vandermonde matrices.

Keywords: Quicksort, average case, Vandermonde.

AMS Subject Classification 2010: 68P10, 68W20.

1 Introduction

The Quicksort algorithm invented by Hoare [7] sorts n keys by randomly choosing a key called pivot and rearranging the array by comparing every key to the pivot, so that all keys less than or equal to the pivot are on its left and all keys greater than or equal to the pivot are on its right. The algorithm is then recursively applied to each of these two smaller arrays (which either might be empty) till we get trivial arrays of length 1 or 0. The term “key” can be a number, word and more generally can be an element of a finite set, equipped with a transitive relation. Throughout this note, we assume that the input array is a random permutation of the positive integers $\{1, \dots, n\}$ with all the $n!$ permutations equally likely to be the input.

A generalisation of the algorithm is to randomly choose k pivots i_1, i_2, \dots, i_k , where $k = 1, 2, \dots$ and partition the array to $(k + 1)$ subarrays. The algorithm is recursively applied to each of the segments that contains at least $(k + 1)$ keys and arrays with less than $(k + 1)$ keys are sorted by another algorithm, as insertion sort. We point out at once that this multipivot Quicksort is a special case of Hennequin’s ‘generalised Quicksort’ [5], where a random sample of $k(t + 1) - 1$ keys is chosen from the array to be sorted and the $(t + 1)$ -st, $2(t + 1)$ -th, \dots , $(k - 1)(t + 1)$ -th smallest keys are used as pivots. Obviously, for $t = 0$, the array is partitioned to k subarrays, according to $k - 1$ pivots. For $k = 2$, we have the ‘median of $(2t + 1)$ ’ Quicksort. For various multipivot variants, we also refer the reader to the Ph.D. theses of Sedgewick [10] and Tan [11].

[‡]E-mail: iliopou@gmail.com

In this note, we consider the average case analysis of multipivot Quicksort and compute the constants of integration by Vandermonde matrices. Let $f(n, k)$ denote the expected cost when a randomly permuted array of n keys is to be sorted by the application of Quicksort on k pivots. We deliberately allow some flexibility in the form of the cost, but a typical example might be the number of comparisons made. We obtain the following recursive relation:

$$f(n, k) = T(n, k) + \frac{1}{\binom{n}{k}} \sum_{\substack{i'_1 \\ i'_1 < i'_2 < \dots < i'_k}} \sum_{i'_2} \dots \sum_{i'_k} \left(f(i'_1 - 1, k) + f(i'_2 - i'_1 - 1, k) + \dots + f(n - i'_k, k) \right),$$

where $\mathbb{E}(\tau(n, k)) = T(n, k) = \bar{a}(k)n + \bar{b}(k)$ is the average value of a “toll function” $\tau(n, k)$ during the first partitioning stage. We assume that this is a linear function of n . The recursion may look a complex k -index summation, but can be simplified by noting the ranges of the indices;

$$\begin{aligned} f(n, k) &= T(n, k) + \frac{1}{\binom{n}{k}} \sum_{i'_1=1}^{n-k+1} \sum_{i'_2=i'_1+1}^{n-k+2} \dots \sum_{i'_k=i'_{k-1}+1}^n \left(f(i'_1 - 1, k) + \dots + f(n - i'_k, k) \right) \\ &= T(n, k) + \frac{(k+1)!}{n(n-1) \dots (n-k+1)} \sum_{i'_1=1}^{n-k+1} \binom{n-i'_1}{k-1} f(i'_1 - 1, k), \end{aligned}$$

since the partitioning of the array according to k pivots yields $(k+1)$ segments and using the fact that the expectations of the average costs in each segment are equal owing to the uniform distribution of the permutation.

2 Solution of the Cauchy-Euler differential equation

With the view of applying generating functions for the solution of our recurrence, let $f(n, k) = a_n$ and consider $h(x) = \sum_{n=0}^{\infty} a_n x^n$;

$$\sum_{n=0}^{\infty} \binom{n}{k} a_n x^n = \sum_{n=0}^{\infty} \binom{n}{k} T(n, k) x^n + (k+1) \sum_{n=0}^{\infty} \left(\sum_{i'_1=1}^n \binom{n-i'_1}{k-1} a_{i'_1-1} \right) x^n.$$

Interchanging the order of summation and multiplying both sides by $\left(\frac{x}{1-x} \right)^{-k}$, this becomes a k -th order differential equation

$$(1-x)^k h^{(k)}(x) = \frac{k! (\bar{a}(k)(x+k) + \bar{b}(k)(1-x))}{(1-x)^2} + h(x)(k+1)!,$$

which is a *Cauchy–Euler* differential equation. This type of differential equations is inherent to the analysis of Quicksort and its variants: we refer the reader to [2], [3], [5, 6] and [10]. Substituting $z = 1 - x$, we have $h(x) = g(1 - x)$ and

$$(-1)^k z^k g^{(k)}(z) - g(z)(k+1)! = \frac{k! (\bar{a}(k)(1 - z + k) + \bar{b}(k)z)}{z^2}.$$

Following the analysis of Hennequin [5, 6] and Sedgewick [10], we use the differential operator Θ , with $\Theta g(z) := zg'(z)$ for the solution of the differential equation. It is easily verifiable by induction that $\binom{\Theta}{k}g(z) = \frac{z^k g^{(k)}(z)}{k!}$ and we have

$$((-1)^k \Theta(\Theta - 1) \dots (\Theta - k + 1) - (k+1)!)g(z) = \frac{k! (\bar{a}(k)(1 - z + k) + \bar{b}(k)z)}{z^2}.$$

The indicial polynomial $\mathcal{P}_k(\Theta)$ is equal to

$$\mathcal{P}_k(\Theta) = (-1)^k \Theta^{\underline{k}} - (k+1)!,$$

where using the notation from [4], $\Theta^{\underline{k}} := \Theta(\Theta - 1) \dots (\Theta - k + 1)$ denotes the falling factorial.

It can be easily proved that the polynomial has k simple roots, with real parts in $[-2, k+1]$. The solution of the differential equation is

$$\begin{aligned} g(z) = & \frac{\bar{a}(k)(k+1)!}{(-2-r_1)(-2-r_2)\dots(-2-r_{k-1})} \frac{\ln(z)}{z^2} \\ & + \frac{k!}{(-1-r_1)(-1-r_2)\dots 1} \frac{(\bar{b}(k) - \bar{a}(k))}{z} + \sum_{i=1}^k s_i z^{r_i}. \end{aligned} \quad (1)$$

In order to evaluate $\mathcal{S}_{k-1}(-2) = (-2-r_1)(-2-r_2)\dots(-2-r_{k-1})$, note that

$$\mathcal{S}_{k-1}(-2) = \mathcal{P}'_k(-2),$$

thus

$$\mathcal{S}_{k-1}(-2) = -(k+1)!(H_{k+1} - 1).$$

Moreover,

$$\mathcal{P}_k(-1) = -kk!$$

and in terms of series, we have

$$\begin{aligned} h(x) = & \frac{\bar{a}(k)}{H_{k+1} - 1} \sum_{n=0}^{\infty} ((n+1)H_n - n)x^n + \sum_{n=0}^{\infty} \sum_{i=1}^k s_i (-1)^n \binom{r_i}{n} x^n \\ & + \frac{\bar{a}(k) - \bar{b}(k)}{k} \sum_{n=0}^{\infty} x^n. \end{aligned} \quad (2)$$

Extracting the coefficients and noting that -2 is the unique root with the least real part, the expected cost of Quicksort on k uniformly at random chosen pivots is

$$a_n = \frac{\bar{a}(k)}{H_{k+1} - 1} ((n+1)H_n - n) + s_k(n+1) + o(n).$$

3 Computation of the integration constants using Vandermonde matrices

In this section, we compute the constants of integration s_i using Vandermonde determinants. We remark that this approach is employed in [3], where the nine integration constants involved in the expected number of comparisons of ‘remedian of 3^2 ’ Quicksort are computed using Vandermonde matrices. In [5, 6], the constant corresponding to the root -2 is computed by the application of generating functions and the differential operator (see Proposition **III.8** in [5, page 50]). Also, Vandermonde determinants appear in the analysis of multiple Quickselect [9]. Our system of equations is

$$g(1) = g'(1) = \dots = g^{(k-1)}(1) = 0.$$

Differentiating m times Eq. (1) and setting $z = 1$,

$$\sum_{i=1}^k s_i r_i^m = (-1)^{m+1} m! \left(\frac{\bar{a}(k)(m+1)H_m - m)}{(H_{k+1} - 1)} + \frac{\bar{a}(k) - \bar{b}(k)}{k} \right), \quad (3)$$

for $m = 0, 1, \dots, k-1$. In matrix form, Eq. (3) is

$$\begin{bmatrix} 1 & 1 & \dots & 1 \\ r_1 & r_2 & \dots & -2 \\ \vdots & \vdots & \ddots & \vdots \\ r_1^{k-1} & r_2^{k-1} & \dots & (-2)^{k-1} \end{bmatrix} \begin{bmatrix} s_1 \\ s_2 \\ \vdots \\ s_k \end{bmatrix} = \begin{bmatrix} -\frac{1}{k}(\bar{a}(k) - \bar{b}(k)) \\ \frac{\bar{a}(k)}{H_{k+1} - 1} + \frac{1}{k}(\bar{a}(k) - \bar{b}(k)) \\ \vdots \\ (-1)^k (k-1)! \left(\frac{\bar{a}(k)(kH_{k-1} - (k-1))}{H_{k+1} - 1} + \frac{\bar{a}(k) - \bar{b}(k)}{k} \right) \end{bmatrix}$$

It is easy to see that the coefficient matrix is non-singular. Using the generating function $x^{k-1} = \sum_{j=1}^{\infty} \left\{ \begin{smallmatrix} k-1 \\ j-1 \end{smallmatrix} \right\} x^{j-1}$ [1], where $\left\{ \begin{smallmatrix} n \\ k \end{smallmatrix} \right\}$ are the Stirling numbers of the second kind, with $\left\{ \begin{smallmatrix} n \\ k \end{smallmatrix} \right\} = 0$ for $n < k$, we can write each power of r_i as a sum of integer multiples of (earlier) rows of the matrix of coefficients we get naturally (and that of course does not change the determinant). Hence the determinant of this matrix is the same as the determinant of the Vandermonde matrix, which is well known to be $\prod_{1 \leq i < j \leq k} (r_j - r_i) \neq 0$, as the roots are all simple.

Transforming the matrix into a Vandermonde one, we have

$$\sum_{j=1}^{\infty} (-1)^j (j-1)! \left\{ \begin{smallmatrix} k-1 \\ j-1 \end{smallmatrix} \right\} \left(\frac{\bar{a}(k)(jH_j - j)}{H_{k+1} - 1} + \frac{\bar{a}(k) - \bar{b}(k)}{k} \right).$$

Note that [1],

$$\sum_{j=1}^k (-1)^j (j-1)! \left\{ \begin{matrix} k-1 \\ j-1 \end{matrix} \right\} = (-1)^k,$$

since $(-1)^{j-1} (j-1)! = (-1)^{\underline{j-1}}$. Also,

$$(-1)^{\underline{j}} = (-1)(-2)^{\underline{j-1}},$$

hence

$$\sum_{j=1}^k (-1)^j j! \left\{ \begin{matrix} k-1 \\ j-1 \end{matrix} \right\} = (-1)^k 2^{k-1}.$$

Differentiating the generating function, we have

$$(k-1)x^{k-2} = \sum_{j=2}^k \left\{ \begin{matrix} k-1 \\ j-1 \end{matrix} \right\} x^{j-1} \left(\sum_{i=0}^{j-2} \frac{1}{x-i} \right),$$

therefore

$$\begin{aligned} (-1)^k (k-1) 2^{k-2} &= \sum_{j=2}^k (-1)^j j! \left\{ \begin{matrix} k-1 \\ j-1 \end{matrix} \right\} \left(\sum_{i=0}^{j-2} \frac{1}{i+2} \right) \\ &= \sum_{j=2}^k (-1)^j j! \left\{ \begin{matrix} k-1 \\ j-1 \end{matrix} \right\} (H_j - 1). \end{aligned} \quad (4)$$

Note that Eq. (4) is a special case of Eq. (36) from [9], for $x = 1$.

Our linear system now becomes:

$$\begin{bmatrix} 1 & 1 & \dots & 1 \\ r_1 & r_2 & \dots & -2 \\ \vdots & \vdots & \ddots & \vdots \\ r_1^{k-1} & r_2^{k-1} & \dots & (-2)^{k-1} \end{bmatrix} \begin{bmatrix} s_1 \\ s_2 \\ \vdots \\ s_k \end{bmatrix} = \begin{bmatrix} -\frac{1}{k}(\bar{a}(k) - \bar{b}(k)) \\ \frac{\bar{a}(k)}{H_{k+1}-1} + \frac{1}{k}(\bar{a}(k) - \bar{b}(k)) \\ \vdots \\ (-1)^k \left((k-1)2^{k-2} \frac{\bar{a}(k)}{H_{k+1}-1} + \frac{\bar{a}(k) - \bar{b}(k)}{k} \right) \end{bmatrix}$$

The inverse matrix can be factored into a product of an upper and lower triangular matrices, [8], [12]. In [8] an algorithm is presented, where the entries of the triangular matrices are recursively computed. Letting \mathbf{A}^{-1} denote the inverse, we

have

$$\mathbf{A}^{-1} = \begin{bmatrix} 1 & \frac{1}{r_1-r_2} & \frac{1}{(r_1-r_2)(r_1-r_3)} & \cdots \\ 0 & \frac{1}{r_2-r_1} & \frac{1}{(r_2-r_1)(r_2-r_3)} & \cdots \\ 0 & 0 & \frac{1}{(r_3-r_1)(r_3-r_2)} & \cdots \\ 0 & 0 & 0 & \cdots \\ \vdots & \vdots & \vdots & \cdots \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & \cdots \\ -r_1 & 1 & 0 & \cdots \\ r_1 r_2 & -(r_1 + r_2) & 1 & \cdots \\ -r_1 r_2 r_3 & r_1 r_2 + r_1 r_3 + r_2 r_3 & -(r_1 + r_2 + r_3) & \cdots \\ \vdots & \vdots & \vdots & \cdots \end{bmatrix}$$

The constants of integration are given by:

$$s_i = (-1)^k \left(\frac{\prod_{j \neq i}^k (r_j + 1)}{\prod_{j \neq i}^k (r_i - r_j)} \left(\frac{\bar{a}(k) - \bar{b}(k)}{k} \right) + \frac{\bar{a}(k)}{H_{k+1} - 1} \frac{\sum_{j=1}^{k-1} \left(\prod_{l \neq j}^{k-1} (r_l + 2) \right)}{\prod_{j \neq i}^k (r_i - r_j)} \right).$$

The products of pairwise differences of roots r_i and r_j that naturally arise in **LU** triangular decomposition of the inverse of Vandermonde matrix form alternating polynomial functions. Putting $i = k$ to the previous equation,

$$\frac{\prod_{j=1}^{k-1} (r_j + 1)}{\prod_{j=1}^{k-1} (-2 - r_j)} = (-1)^{k-1} \frac{\mathcal{P}_k(-1)}{\mathcal{S}_{k-1}(-2)} = (-1)^{k-1} \frac{k}{(k+1)(H_{k+1} - 1)}$$

and the sum of the products is

$$\sum_{j=1}^{k-1} \left(\prod_{l \neq j}^{k-1} (r_l + 2) \right) = \mathcal{S}'_{k-1}(-2).$$

Differentiating $\mathcal{P}_k(\Theta)$ twice and setting $\Theta = -2$,

$$\mathcal{S}'_{k-1}(-2) = \frac{\mathcal{P}_k''(-2)}{2} = \frac{(k+1)!(H_{k+1}^2 - 2H_{k+1} - H_{k+1}^{(2)} + 2)}{2},$$

where $H_{k+1}^{(2)} := \sum_{j=1}^{k+1} \frac{1}{j^2}$ denotes the second-order harmonic number.

The main result of this paper is the following Theorem:

Theorem 3.1. *The expected cost of multipivot Quicksort on k uniformly at random selected pivots for partitioning an array consisting of $n > k$ distinct keys to subarrays that each one contains at most k keys is*

$$\begin{aligned} & \frac{\bar{a}(k)}{H_{k+1} - 1} ((n+1)H_n - n) - \left(\frac{\bar{a}(k)}{H_{k+1} - 1} \left(\frac{H_{k+1}^2 - 2H_{k+1} - H_{k+1}^{(2)} + 2}{2(H_{k+1} - 1)} \right) \right. \\ & \left. + \frac{\bar{a}(k) - \bar{b}(k)}{(k+1)(H_{k+1} - 1)} \right) (n+1) + o(n), \end{aligned}$$

where the “toll function” has the average value $\bar{a}(k)n + \bar{b}(k)$.

Acknowledgements

I thank Dr. David B. Penman for his advice and helpful suggestions regarding this paper.

References

- [1] M. Abramowitz and I. A. Stegun, *Handbook of Mathematical Functions with Formulas, Graphs, and Mathematical Tables*. Dover Publications, 1972.
- [2] H. H. Chern, H. K. Hwang and T. H. Tsai, *An asymptotic theory for Cauchy–Euler differential equations with applications to the analysis of algorithms*. J. Algorithm. **44**, 177-225, 2002.
- [3] M. Durand, *Asymptotic analysis of an optimized Quicksort algorithm*. Inform. Proc. Lett. **85** (2): 73-77, 2003.
- [4] R. Graham, D. E. Knuth and O. Patashnik, *Concrete Mathematics: A Foundation for Computer Science*. Addison–Wesley Publishing, 2nd Edition, 1994.
- [5] P. Hennequin, *Analyse en moyenne d’algorithmes: tri rapide et arbres de recherche*. Ph.D. thesis. École Polytechnique, 1991.
- [6] P. Hennequin, *Combinatorial analysis of quicksort algorithm*. RAIRO Theor. Inform. Appl. **23** (3): 317-333, 1989.
- [7] C. A. R. Hoare, *Quicksort*. Comput. J. **5** (1): 10-15, 1962.
- [8] S. H. Hou and E. Hou, *Triangular Factors of the Inverse of Vandermonde Matrices*. Proceedings of the International MultiConference of Engineers and Computer Scientists. Vol. II IMECS, Hong Kong, 2008.
- [9] A. Panholzer, *Analysis of multiple quickselect variants*. Theor. Comput. Sci. **302** (1-3): 45-91, 2003.
- [10] R. Sedgewick, *Quicksort*. Ph.D. thesis. Garland Publishing, 1980.
- [11] K. H. Tan, *An asymptotic analysis of the number of comparisons in multipartition quicksort*. Ph.D. thesis. Carnegie Mellon University, 1993.
- [12] L. R. Turner, *Inverse of the Vandemonde Matrix with Applications*. National Aeronautics and Space Administration, technical note D-3547, 1966.